UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/052,250	01/23/2002	Russell T. Davis	7643.0042	1920

22852 7590 11/01/2007
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER
LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER
NGUYEN, CHAU T

ART UNIT	PAPER NUMBER
2176	

MAIL DATE	DELIVERY MODE
11/01/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/052,250	**Applicant(s)** DAVIS ET AL.	
	Examiner Chau Nguyen	**Art Unit** 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE ____ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 16 August 2007.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-6,8-21,23-34 and 36-64 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-6,8-21,23-34 and 36-64 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date 6/19/2007 & 10/01/2007.

4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. Applicant's submission filed on 08/16/2007 has been entered. Claims 1-6, 8-21, 23-34, and 36-64 are presented for examination.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 62-64 are rejected under 35 U.S.C. 103(a) as being unpatentable over Krug et al. (Krug), US Patent No. 6,721,736 and further in view of further in view of Hamscher et al. (Hamscher), published on 07/31/2000.

4. As to claim 62, Krug discloses a data processing comprising:

a parser that (col. 8, lines 21-30: a syntax tree parser):

receives one or more text documents (col. 7, lines 46-50, col. 8, lines 21-30, and Fig. 3: HTML document is inputted to the interface 8)

interprets tags included in the one or more text documents to create software elements (col. 8, lines 21-30: the syntax tree parser analyses the HTML document by recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements (software elements), and

determines the hierarchy of the software elements within a structure representative of the one or more text documents (col. 8, lines 21-30: recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements).

However, Krug does not explicitly disclose a manager that provides for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document.

Hamscher discloses XBRL consisting of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances, and taxonomies can be composed together to extend other taxonomies (page 1, Abstract and page 6). Hamscher also discloses combining of financial information from different periods or entities or even for the same entity under different reporting regimes, and creating an XBRL instance document (hierarchy) by concatenating other XBRL instance documents (page 17, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Krug to include providing for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document. Hamscher suggests that XBRL allows software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

5. As to claim 63, Krug discloses the structure is a Numerator Document Object Model (NDOM) (col. 7, lines 46-50, col. 8, lines 21-30, and Fig. 3).

6. As to claim 64, Krug disclose a parser that receives text documents, interprets tags of the text document, and determines the hierarchy of the software elements within the text document as discussed in claims 62-63 above.

However, Krug does not explicitly disclose wherein the one or more text documents are XBRL documents.

Hamscher discloses XBRL consists of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances (pages 1-3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Krug to include XBRL documents because XBRL documents allow software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

7. Claims 1-6, 11-21, 24-33, 34, 37-46, 49-57, and 59-61 are rejected under 35 U.S.C. 103(a) as being unpatentable over Saxton, US Patent No. 6,370,549 in view of Polizzi et al. (Polizzi), US Patent Application Publication No. 2002/0052954 and further in view of Hamscher et al. (Hamscher), published on 07/31/2000.

8. As to claims 1, 17, 29, 30, 42, and 54 Saxton discloses a data processing system for developing reports, comprising:

a parser that receives one or more text documents and creates software elements having a format with a hierarchal relationship between the software elements based on the one or more text documents (Abstract, col. 5, lines 63 – col. 6, lines 16); and

However, Saxton does not explicitly disclose an editor that develops a report by referencing the software elements created from the one or more text documents to form a structure of the report and retrieves data from one or more sources to represent one or more values within the report and wherein a mapper generates a relationship between the data from the one or more sources and the one or more values to be placed within the report.

Polizzi discloses in Abstract, page 3, paragraph [0024] and page 6, paragraph [0039]: a repository stores all computer files, which are called objects, and the objects can be any computer file such as text documents; these text documents are organized or arranged in a hierarchy). Polizzi also discloses preparing a report based upon retrieved data (Abstract), thus a mapper must be inherent from Polizzi's system since it prepared a reports based upon retrieved data (a relationship between the data from the one or more sources and the one or more values to be placed within the report).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Polizzi and Saxton to include develops reports by referencing the software elements created from the one or more text documents and retrieves data from one or more sources to represent one or more values within the report and wherein a mapper generates a relationship between the data from the one or more sources and the one or more values to be placed within the report in order to allow user to view and select reports.

However, Saxton and Polizzi do not explicitly disclose a manager that provides for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document.

Hamscher discloses XBRL consisting of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances, and taxonomies can be composed together to extend other taxonomies (page 1, Abstract and page 6). Hamscher also discloses combining of financial information form different periods or entities or even for the same entity under different reporting regimes, and creating an XBRL instance document (hierarchy) by concatenating other XBRL instance documents (page 17, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher with Saxton and Polizzi to include providing for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document. Hamscher suggests that XBRL allows software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

9. As to claims 2, 18, 31, 43, and 55, Saxton and Polizzi (Saxton-Polizzi) disclose wherein the format with the hierarchal relationship between the software elements is a Numerator Document Object Model (NDOM) (Saxton, Abstract).

10. As to claims 3, 19, 32, 44, and 56, Saxton-Polizzi, however, do not disclose wherein the one or more text documents are XBRL documents.

Hamscher discloses XBRL consists of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances (pages 1-3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Saxton-Polizzi to include XBRL documents because XBRL documents allow software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

11. As to claims 4, 20, 33, 45, and 57, Saxton-Polizzi, however, do not disclose wherein the parser creates the software elements having the format with the hierarchal relationship by interpreting tags included in the one or more text documents.

Hamscher discloses in pages 1-3 and 7 that XBRL consists of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances (pages 1-3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Saxton-Polizzi to include XBRL documents because XBRL documents allow software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

12. As to claim 5, Saxton-Polizzi disclose wherein a manager manipulates the software elements (Saxton, col. 5, line 63 – col. 6, line 65).

13. As to claims 6, 21, 34, and 46, Saxton-Polizzi disclose wherein the manager manipulates the software elements by browsing, editing, loading, and storing the

software elements (Saxton, col. 6, line 66 – col. 7, line 11; Polizzi, Abstract, and page 1, paragraphs [0005]-[0006]).

14. As to claim 11, Saxton discloses a parser that receives one or more text documents and creates software elements having a format with a hierarchal relationship between the software elements based on the one or more text documents as discussed in claims 1, 17, 29, 30, 42, and 54 above.

However, Saxton does not disclose wherein a mapper links the report and the one or more sources that will present one or more values within the report.

Polizzi discloses in the Abstract, pages 3-4, paragraphs [0024]-[0026], and page 6, paragraph [0039]: a repository stores all computer files, which are called objects, and the objects can be any computer file such as text documents; these text documents are organized or arranged in a hierarchy). Polizzi also discloses preparing a report based upon retrieved data (Abstract).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Polizzi and Saxton to include develops reports by referencing the software elements created from the one or more text documents and retrieves data from one or more sources to represent one or more values within the report in order to allow user to view and select reports.

15. As to claims 12, 24, 37 and 49, Saxton-Polizzi disclose wherein the report and the one or more sources are linked through a "drag and drop" process (Saxton, col. 1, line 61 – col. 2, line 12 and col. 7, lines 12-24).

16. As to claims 13, 25, 38, 50, and 59, Saxton discloses a parser that receives one or more text documents and creates software elements having a format with a hierarchal relationship between the software elements based on the one or more text documents as discussed in claims 1, 17, 29, 30, 42, and 54 above.

However, Saxton does not explicitly disclose wherein the editor provides for the software elements to be modified to create a new combination of software elements representative of a new text document.

Polizzi discloses object or category within the repository is modified (page 4, paragraph [0027]).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Polizzi and Saxton to include modified object or category in order to provide a copy of the output report as an automatic update to a user's page.

17. As to claims 14, 26, 39, 51, and 60, Saxton discloses a parser that receives one or more text documents and creates software elements having a format with a hierarchal relationship between the software elements based on the one or more text documents as discussed in claims 1, 17, 29, 30, 42, and 54 above.

However, Saxton does not explicitly disclose wherein the editor provides for modification of one or more parameters associated with the software elements.

Polizzi discloses object or category within the repository is modified (page 4, paragraph [0027].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Polizzi and Saxton to include modified object or category in order to provide a copy of the output report as an automatic update to a user's page.

18. As to claims 15, 27, 40, 52 and 61, Saxton discloses a parser that receives one or more text documents and creates software elements having a format with a hierarchal relationship between the software elements based on the one or more text documents as discussed in claims 1, 17, 29, 30, 42, and 54 above.

However, Saxton does not explicitly disclose wherein the software elements are transformed to new software elements and are imported into an RDL system.

Polizzi discloses in the Abstract, page 2, paragraphs [0008] and [0020]: the portal page is an object arranged in a format that is readable by a browser program, and the user interface may be based upon a standard browser program that is capable of reading HTML, Java, XML, or other languages.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Polizzi and Saxton to include using a standardized browser program as a user interface so the user can point and click on hypertext links to navigate through the portal system which provides the ability to search both structured and unstructured data.

19. As to claims 16, 28, 41, and 53, Saxton-Polizzi disclose wherein the software elements are transformed to the new software elements by retrieving a tag associated with each of the software elements in a dictionary and invoking a translation routine associated with the tag (Saxton, col. 7, line 44 – col. 10, line 30).

20. Claims 8-10, 23, 36, 47-48, and 58 are rejected under 35 U.S.C. 103(a) as being unpatentable over Saxton, Polizzi et al. (Polizzi) and Hamscher as applied to claims 1-6, 11-21, 24-33, 34, 37-46, 49-57, and 59-61 above, and further in view of Clancey et al. (Clancey), US Patent No. 6,134,563.

21. As to claim 8 and 47, Saxton-Polizzi-Hamscher, however, do not explicitly disclose wherein one or more templates are used to develop the report, which contain data that is directly inserted into the report and instructions enabling data from the one or more source to be inserted into the report.

Clancey discloses a user can create and edit a report, which is created based upon a predefined template (col. 24, lines 5-30).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Clancey and Saxton-Polizzi-Hamscher to include using templates to develop the report which contain data that is directly inserted into the report and instructions enabling data from the one or more source to be inserted into the report. By using templates to develop reports, it would provide user-friendly environment and save time for users.

22. As to claims 9, 23, 36, and 58, Saxton-Polizzi-Hamscher, however, do not explicitly disclose wherein the one or more templates contain data that is directly inserted into the report and instructions enabling data from the one or more sources to be inserted into the report.

Clancey discloses a user can create and edit a report, which is created based upon a predefined template (col. 24, lines 5-30).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Clancey and Saxton-Polizzi-Hamscher to include using templates to develop the report. By using templates to develop reports, it would provide user-friendly environment and save time for users.

23. As to claims 10 and 48, Saxton-Polizzi-Hamscher, however, do not explicitly disclose wherein the one or more templates provide instructions to a mapper to retrieve the data that is directly inserted into the report and data from local or remote sources.

Clancey discloses a user can create and edit a report, which is created based upon a predefined template (col. 24, lines 5-30).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Clancey and Saxton-Polizzi-Hamscher to include using templates to develop the report. By using templates to develop reports, it would provide user-friendly environment and save time for users.

Response to Arguments

In the remarks, Applicant(s) argued in substance that

A) "A prima facie case of obviousness has not been established because, among other things, Krug nor Hamscher, taken alone or in any proper combination, fail to teach or suggest each and every element recited by Applicants' claim (claim 62)." (see page 3 of the remarks).

In reply to argument A, to establish a prima facie case of obviousness, three basic criteria must be met.

First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. In this case, Krug discloses recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements (software elements), which is similar to XBRL consisting of XML elements and attributes of Hamscher, thus it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Krug to include XBRL documents because XBRL documents allow software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

Second, there must be a reasonable expectation of success. The prior art can be modified or combined to reject claims as prima facie obvious as long as there is a reasonable expectation of success. In re Merck & Co., Inc., 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). In this case, claimed invention directed to a method for data processing comprising a parser for receiving text documents which are XBRL documents, interpreting tags of the documents and determining the hierarchy of the software elements of the documents, was rejected as obvious over a reference (Krug) which taught recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements (software elements) and further in view of Hamscher reference which taught XBRL consisting of XML elements and attributes. Thus, there was reasonable expectation that a process combining the prior art steps could be successfully scaled up.

Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. In this case, Krug discloses a data processing comprising:

a parser that (col. 8, lines 21-30: a syntax tree parser):

receives one or more text documents (col. 7, lines 46-50, col. 8, lines 21-30, and Fig. 3: HTML document is inputted to the interface 8)

interprets tags included in the one or more text documents to create software elements (col. 8, lines 21-30: the syntax tree parser analyses the HTML document by recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements (software elements), and

determines the hierarchy of the software elements within a structure representative of the one or more text documents (col. 8, lines 21-30: recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements).

However, Krug does not explicitly disclose a manager that provides for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document.

Hamscher discloses XBRL consisting of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements referred to in document instances, and taxonomies can be composed together to extend other taxonomies (page 1, Abstract and page 6). Hamscher also discloses combining of financial information form different periods or entities or even for the same entity under different reporting regimes, and creating an XBRL instance document (hierarchy) by concatenating other XBRL instance documents (page 17, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Hamscher and Krug to include providing for the creation of a second hierarchy of the software elements and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document. Hamscher suggests that XBRL allows software vendors, programmers and end users who adopt it as a specification to enhance the creation, exchange, and comparison of business reporting information.

B) "Hamscher is silent regarding a "hierarchy"" and "Hamscher does not "determine the hierarchy of software elements" and create a first and second hierarchy" (see page 4 of the Remarks).

In reply to argument B, In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). In this case Hamscher discloses "XBRL consists of a core language of XML elements and attributes used in document instances as well as a language used to define new elements and taxonomies of elements (creating hierarchies) referred to in document instances (page 1, Abstract). Hamscher also discloses combining of financial information from different periods or entities or even for the same entity under different reporting regimes, and creating an XBRL instance document (hierarchy) by concatenating other XBRL instance documents (page 17, 1st paragraph).

Krug discloses col. 8, lines 21-30: the syntax tree parser analyses the HTML document by recognizing the HTML tags within the document and constructing a hierarchical HTML syntax tree that represents the hierarchical relationship of the syntax elements (software elements).

C) "The XBRL instance document is the result of concatenating other XBRL instance documents. This resulting XBRL instance document is not "restructured."" (see page 4 of the Remarks).

In reply to argument C, by definition of "restructure" is to construct or form anew or provide with a new structure. Since the XBRL instance document is the result of concatenating other XBRL instance documents, therefore, the XBRL instance document is "restructured".

D) Hamscher does not teach or suggest a manager that "provide for creation of a second hierarchy between the software elements, and provides for the restructuring of the first hierarchy and the second hierarchy into software structures corresponding to a new text document (see page 5 of the Remarks).

In reply to argument D, this argument is similar to arguments B, which is already addressed above.

E) Clancey does not teach or suggest "a manager that provides for the creation of a second hierarchical relationship between the software elements, and the restructuring of the first hierarchical relationship and the second hierarchical relationship into software structures corresponding to a new text document." (see page 7 of the Remarks)

In reply to argument E, the examiner does not use Clancey reference to reject the limitation argued in argument E. Instead, the examiner uses Hamscher reference to reject this limitation in which it is discussed above in claims 1 and 62.

24. Applicant's arguments filed 08/16/2007 have been fully considered but they are not persuasive. Please see the response to arguments and the rejection above.

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chau Nguyen whose telephone number is (571) 272-4092. The examiner can normally be reached on 8:30 am – 5:30 pm Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton, can be reached on (571) 272-4137. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306. On July 15, 2005, the Central Facsimile (FAX) Number will change from 703-872-9306 to 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chau Nguyen
Patent Examiner
Art Unit 2176

/Doug Hutton/
Doug Hutton
Supervisory Primary Examiner
Technology Center 2100